

IOT Based Dipstick Type Engine Oil Level and Impurities Monitoring System A Portable Online Spectrophotometer

B. Sandeep¹, D. Hasritha, K². Rupesh Chowdary³, S. Priyanka⁴, T.N.V. Raga Supriya⁵

Sr. Assistant Professor¹, Student², Student³, Student⁴, Student⁵

ELECTRONICS AND INSTRUMENTATION ENGINEERING,

¹LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING, Mylavaram, India

Abstract - This project involves developing an IoT-based instrument for monitoring oil impurities and levels in internal combustion engine sumps. The method used is customized white LED spectrophotometry for qualitative and quantitative analysis of lubricant oil. The system uses a light-dependent sensor with a sensitivity of less than 90% and a range of 500-600nm. The instrument is affordable, portable, and includes a mobile application for remote monitoring. This project is a significant contribution to the field of engine maintenance and presents a compelling case for the implementation of IoT-based monitoring systems in the maintenance of internal combustion engines. keywords for above paragraph

Index Terms - IoT-based instrument, oil impurities, internal combustion engine sumps, customized white LED spectrophotometry, qualitative analysis, quantitative analysis, light-dependent sensor, sensitivity, affordability, portability, mobile application, remote monitoring, engine maintenance.

Introduction

A novel technique enables real-time monitoring of engine oil quality utilizing a portable online spectrophotometer and an IOT based dipstick type engine oil level and impurity monitoring device. For the dependable and secure operation of machinery, this system offers a simple and affordable option for engine oil health monitoring. The device measures the oil level and looks for pollutants using a dipstick-style sensor that is put into the engine oil reservoir. A spectrophotometer that is part of the sensor examines the oil's colour to assess its quality. Wireless transmission of the sensor's data to a cloud-based platform allows for real-time analysis and visualization of the data.

II. LITERATURE SURVEY

An intriguing field of study is a portable online spectrophotometer-based dipstick type engine oil level and contaminants monitoring device for the Internet of Things (IoT). The objective is to create a portable instrument that uses a spectrophotometer to assess the oil level and contaminants in an engine in real-time. Dangling Ling, ET AL., "A dipstick-based colorimetric test for evaluating oil quality using smart phone." The authors of this research suggested a dipstick-based colorimetric test for smart phone-based engine oil quality monitoring. The nitrocellulose membrane used to make the dipstick was covered with an assortment of organic dyes that changed color when in contact with the oil. A subsequent colour shift was noticed by a specially created programmer was used to evaluate the data from a smart phone camera. Wang Xiaoping, ET AL "Wireless sensor network-based oil condition monitoring system for engines." The authors of this research suggested an engine oil condition monitoring system based on wireless sensor networks. A sensor node, a gateway, and a cloud server made up the system. The oil condition parameters, including oil level, temperature, and contaminants, are measured by the sensor node. The gateway and cloud server were used to transmit the data the sensor node had acquired for analysis. G. Amaranth, ET AL "Design of an IOT based engine oil condition monitoring system." An IOT based system for monitoring engine oil condition was suggested by the authors of this research. Sensors were used in the system's design to detect the oil level, temperature, and contaminants. The micro controller analyzed the data once it was received from the sensors and forwarded it to a cloud-based server for analysis. The outcomes were then shown on a dashboard for immediate observation.

III. COMPONENTS

3.1 ARDUINO

A software development environment and programmable microcontroller make up the open-source Arduino electronics platform, which may be used to write, build, and upload code to the microcontroller. For professionals, artists, and enthusiasts who wish to build interactive projects, it was made to be simple to use.

The Arduino platform comprises of a variety of boards with diverse capabilities and features. The Arduino Uno, Arduino Mega, and Arduino Nano are a few of the often used boards. These boards feature various memory capacities, clock rates, and pin arrangements. Based on C and C++, the programming language used to create code for Arduino boards. The user-friendly interface of the Arduino software development environment makes it easier to write and upload code.



FIG : 3.1.1 ARDUINO

ROBOTICS, HOME AUTOMATION, ENVIRONMENTAL MONITORING, AND A HOST OF ADDITIONAL USES ARE ALL POSSIBLE WITH ARDUINO BOARDS. SINCE THE PLATFORM IS OPEN-SOURCE, A SIZABLE USER COMMUNITY HAS GROWN UP AROUND IT, SHARING KNOWLEDGE, RESOURCES, AND IDEAS. THIS MAKES IT SIMPLER FOR NEW USERS TO UNDERSTAND THE BASICS OF THE PLATFORM AND GET UP AND RUNNING.

OVERALL, ARDUINO IS A STRONG AND ADAPTABLE PLATFORM THAT MAKES IT SIMPLE FOR USERS TO BUILD NEW AND INTERACTIVE PROJECTS.

3.3 ARDUINO UNO:

Arduino Uno is a microcontroller board that utilizes the ATmega328P microcontroller. It comes equipped with a 16 MHz quartz crystal, 14 digital input/output pins, 6 analog inputs, a USB connection, a power jack, an ICSP header, and a reset button. The board can be powered through either a USB connection or an external power supply, providing flexibility in powering the board. The digital input/output pins can be configured as either inputs or outputs, allowing the board to interact with various external devices. The analog inputs can be used to read values from sensors or other analog devices, providing the board with the ability to interact with the physical world.

The USB connection allows for easy communication between the board and a computer, enabling programming and data transfer. The power jack allows for external power supply, providing more power than the USB connection. The ICSP header allows for in-circuit programming of the microcontroller, while the reset button can be used to reset the board.

Overall, Arduino Uno is a versatile and powerful microcontroller board that is well-suited for a wide range of applications. Its flexibility, ease of use, and wide availability of documentation and support make it a popular choice among hobbyists, students, and professionals alike.

Arduino Uno is a popular board among hobbyists and professionals for building various projects such as robots, home automation systems, and electronic gadgets.

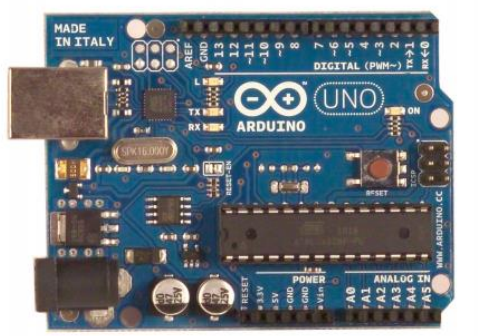


Fig : 3.3.1 Arduino uno

The board is easy to use and program using the Arduino IDE, which is a free open-source software development environment for Arduino boards.

Arduino Uno is also highly customizable, with a wide range of shields and modules available that can be easily connected to the board to expand its capabilities. Overall, the Arduino Uno is a versatile and user-friendly microcontroller board that is ideal for beginners and experienced users alike.

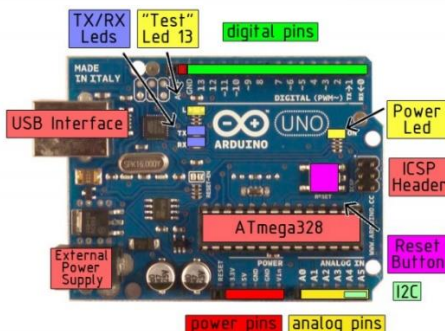


FIG : 3.3.2 ARDUINO UNO

3.4 POWER :

Makers, hobbyists, and professionals all use the well-liked open-source Arduino hardware and software platform to create a variety of electronic projects. The strength of Arduino resides in its adaptability, simplicity, and low cost.

The following are some significant ways that Arduino gives its users power:

Simple to use: Even for individuals with little or no programming knowledge, Arduino is designed to be simple to use. Users may quickly and simply construct and customise electronic creations thanks to its simple, intuitive programming language and visual interface.

Flexibility: From straightforward LED displays to intricate robots and automation systems, Arduino is incredibly versatile and can be utilised for a wide range of projects. Also, because it's open-source, users can tweak it and adapt the platform to their own requirements.

Accessibility: A wide range of users can utilise Arduino because the boards and components are reasonably priced. Because to its accessibility, large-scale projects involving several boards and sensors can be made without breaking the bank.

Community: The active and vast Arduino community offers users a plethora of information and assistance. This comprises tutorials, forums, pre-written code libraries, and other tools that can assist users in problem-solving and skill acquisition.

Ultimately, the strength of Arduino rests in its capacity to enable users to rapidly, conveniently, and economically develop and customise electronic products, giving it the perfect platform for both novice and seasoned makers.

3.5 MEMORY:

The Atmega328 microcontroller contains 32 KB of flash memory, which is used to store code. However, 0.5 KB of this memory is used for the bootloader, which is responsible for initiating the program stored in the flash memory.

In addition to flash memory, the Atmega328 contains 2 KB of SRAM (Static Random Access Memory). SRAM is a type of memory that stores data temporarily and is typically used for storing variables or other data that needs to be accessed quickly by the microcontroller.

The Atmega328 also includes 1 KB of EEPROM (Electrically Erasable Programmable Read-Only Memory). EEPROM is a type of memory that can be programmed and erased electrically, and it retains its contents even when power is turned off. This makes it suitable for storing data that needs to be preserved between power cycles, such as configuration data or calibration values.

Finally, the Atmega328 also contains an additional 1 KB of EEPROM memory, which can be read and written using the EEPROM library. This provides additional flexibility for storing and retrieving data in non-volatile memory.

Overall, the Atmega328 microcontroller provides a significant amount of memory for storing code, data, and other program-related information. The combination of flash memory, SRAM, and EEPROM make it a versatile and powerful microcontroller that is well-suited for a wide range of applications.

3.6 INPUT/OUTPUT :

The Arduino Uno provides 14 digital input/output pins that can be used for various purposes, such as controlling LEDs or reading input from sensors. Additionally, some pins have specialized functions, such as the serial communication pins (RX and TX) used for transmitting and receiving data, and external interrupt pins (2 and 3) that can be programmed to trigger an interrupt based on a specific event. It is important to note that the internal pull-up resistors are disconnected by default, but can be enabled using the `pinMode()` function. Also, all pins operate on 5 volts.

3.7 COMMUNICATION:

The Arduino Uno allows for communication with a computer or other devices using UART TTL serial connection provided by the ATmega328 on digital pins 0 (RX) and 1 (TX). This serial communication is then routed over USB by the ATmega8U2 on the board, which makes it appear as a virtual COM port to the computer. The firmware for the '8U2 uses the USB COM drivers that are already built-in, so there is no need to install any additional drivers. This makes it easy to connect and communicate with the Arduino Uno using just a USB cable.

3.8 USB OVER CURRENT PROTECTION :

The poly fuse is a safety feature that protects the USB ports from overcurrent or short circuits that can cause damage to the computer or the Arduino board. If the current exceeds the safe limit of 500 mA, the poly fuse will trip and disconnect the circuit, preventing any further damage. Once the short or overload is resolved, the poly fuse will automatically reset and the connection can be resumed.

3.9 PHYSICAL CHARACTERISTICS :

The physical dimensions of the Arduino Uno board are important to consider for projects where space is a concern. The board measures 2.7 inches in length and 2.1 inches in width, which are the maximum dimensions of the PCB itself. However, the USB connector and power jack protrude beyond these dimensions.

To secure the board to a surface or container, there are four screw holes available. Additionally, it's worth noting that digital pins 7 and 8 are spaced 160 mil (0.16") apart, which is not an even multiple of the other pins' 100 mil spacing. This may affect the placement of components on a breadboard or PCB design.

3.10 LDR MODULE :**FIG : 3.10.1 LDR MODULE****3.11 OUTPUT TYPE:**

It seems that you are referring to an LDR module in your previous message. The LDR module usually consists of a Light-Dependent Resistor (LDR) and a comparator circuit, which may include an LM393 chip.

The analogue voltage output from the LDR module is typically connected to the analog input (A0) of the Arduino Uno. By reading the analog value from A0 using the `analogRead()` function, you can obtain information about the light intensity detected by the LDR.

In addition to the analog output, some LDR modules may also provide digital outputs (0 and 1) based on a threshold set by a potentiometer on the module. This digital output can be connected to any of the digital pins on the Arduino Uno and can be read using the `digitalRead()` function.

The LDR module is usually designed to be easy to install, with a fixed bolt hole for mounting and power and digital output indicators (red and green LEDs) on the module itself. The LM393 comparator chip is commonly used in these modules to compare the analog voltage from the LDR with a reference voltage set by the potentiometer and to generate the digital output based on the result of the comparison.

3.12 PIN OUTS:

IT SEEMS LIKE THE TEXT YOU PROVIDED IS INCOMPLETE AND MAY BE MISSING SOME CONTEXT. HOWEVER, BASED ON THE INFORMATION PROVIDED, IT APPEARS TO BE DESCRIBING A SMALL BOARD WITH DIGITAL AND ANALOG OUTPUTS. THE BOARD HAS A DIGITAL OUTPUT INTERFACE, LABELED AS "DO," AS WELL AS AN ANALOG OUTPUT INTERFACE, LABELED AS "AO." ADDITIONALLY, THE BOARD HAS EXTERNAL CONNECTIONS FOR VCC (POWER) AND GND (GROUND). THE VCC CONNECTION ALLOWS FOR AN EXTERNAL POWER SUPPLY TO BE USED WITH A VOLTAGE RANGE OF 3.3V-5V. THE PURPOSE AND FUNCTION OF THE BOARD ARE NOT EXPLICITLY STATED, BUT IT APPEARS TO BE DESIGNED FOR USE IN ELECTRONIC PROJECTS REQUIRING BOTH DIGITAL AND ANALOG OUTPUTS.

3.13 LCD :**Fig : 3.13.1 LCD**

LCDs are commonly used in devices such as calculators, watches, and computer monitors due to their low power consumption and ability to display information without the need for backlighting. They can display text, numbers, and even simple graphics.

LCD technology is widely used in televisions, computer monitors, mobile devices, and other types of displays. LCDs are known for their low power consumption, high resolution, and thin design. However, they can suffer from limited viewing angles and slower refresh rates compared to other display technologies like organic light-emitting diodes (OLEDs).

3.14 BUZZER PIN CONFIGURATION

PIN NUMBER	PIN NAME	DESCRIPTION
1	POSITIVE	THE EXTENDED TERMINAL LEADS OR THE (+) SYMBOL SERVE AS INDICATORS FOR THE POLARITY OF THE POWER SOURCE. THIS IS IMPORTANT TO ENSURE THAT THE CORRECT VOLTAGE IS APPLIED TO THE LCD, AS REVERSE POLARITY CAN CAUSE DAMAGE. THE LCD IS TYPICALLY POWERED BY A 6V DC SOURCE, ALTHOUGH THE EXACT VOLTAGE MAY VARY DEPENDING ON THE SPECIFIC MODEL AND APPLICATION. IT IS IMPORTANT TO USE THE CORRECT VOLTAGE TO PREVENT DAMAGE TO THE LCD AND ENSURE PROPER FUNCTIONALITY.
2	NEGATIVE	THE SHORT TERMINATION LEAD IS USED TO IDENTIFY THE COMMON CONNECTION TO THE CIRCUIT'S GROUND. THIS LEAD IS TYPICALLY SHORTER THAN THE OTHERS AND MAY ALSO BE MARKED WITH A SYMBOL SUCH AS A HORIZONTAL LINE OR A SERIES OF VERTICAL LINES TO INDICATE ITS FUNCTION. THE GROUND CONNECTION IS AN ESSENTIAL COMPONENT OF MOST ELECTRONIC CIRCUITS, PROVIDING A REFERENCE POINT FOR THE VOLTAGES AND CURRENTS WITHIN THE CIRCUIT. IT IS TYPICALLY CONNECTED TO THE NEGATIVE TERMINAL OF THE POWER SUPPLY AND SERVES AS A RETURN PATH FOR THE CURRENT FLOWING THROUGH THE CIRCUIT. BY PROVIDING A STABLE REFERENCE VOLTAGE, THE GROUND CONNECTION HELPS TO ENSURE THAT THE CIRCUIT OPERATES CORRECTLY AND WITHOUT INTERFERENCE FROM EXTERNAL SOURCES.

3.15 BUZZER FEATURES AND SPECIFICATIONS :

- The rated voltage of the device is 6V DC, while the operating voltage range is between 4V and 8V DC.
- Constant Beep Sound Type • Maximum current: 30 Ma • Resonant frequency: 2300 Hz.
- Minimal, neatly packaged item.
- Compatible with Perf boards and breadboards.

3.16 HOW TO USE A BUZZER :

A buzzer is a commonly used and straightforward component that can be used to add sound features to a project or system. Its small size and convenient 2-pin structure make it easy to integrate onto PCBs, Perf Boards, and other surfaces, which is why it is used in a wide range of electrical applications.

Buzzers are readily available and fall into two different groups. The buzzer being shown here is a simple gadget that, when turned on, produces a constant bewEEP sound. Another type is a pre-made buzzer that is larger than this and beeps. Beep. Beep. It makes sound because of an oscillating circuit integrated in. The one shown below, however, is the most common since it can be altered by adding other circuits to the pulsating internal circuit

it has the most common, however, may be adjusted with the help of additional circuits to suit our purpose readily, therefore this one is shown below.

3.17 APPLICATIONS OF BUZZER :

- COMMUNICATION TOOLS THAT DEMAND THAT THE USER BE ALERTED TO ANYTHING.
- ALARM CIRCUITRY.
- AUTO-RELATED ELECTRONICS.
- PORTABLE DEVICES BECAUSE OF THEIR COMPACT SIZE.

3.18 WIFI MODULE (ESP8266) :

The ESP8266 was created by the Chinese company Expressive Applications to provide solutions for the Internet of Things (IoT). This chip is a complete Wi-Fi system on a chip, featuring a 32-bit processor, some RAM, and between 512KB and 4MB of flash memory, depending on the source. The ESP8266 can operate as a standalone device that can run a simple programmer or as a wireless adapter that can bring Wi-Fi capability to other systems, making it a versatile and useful component for IoT applications.

3.19 OVERVIEW AND SPECIFICATION :

The ESP8266 module is available from both Expressif and non-Expressif vendors, including Oilex and NodeMCU, in several different versions, ranging from the ESP-01 to the ESP-12. These modules vary in size and additional PCB components, with some versions featuring an integrated PCB antenna and up to 7 GPIO pins, while others lack an antenna and offer more limited access to GPIO but are also less expensive and available in varying module sizes.

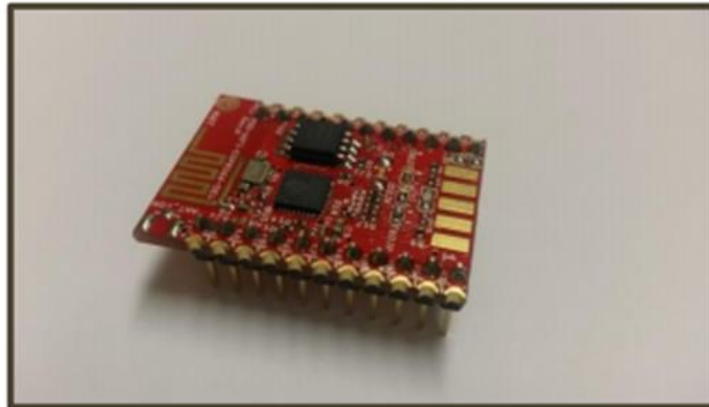


Figure 3.8 Olimex MOD-WIFI-ESP8266-DEV module used in this thesis

FIG : 3.19.1 OVERVIEW AND SPECIFICATION

THE OILEX MOD-WIFI-ESP8266-DEV MODULE USED IN THIS THESIS INCLUDES ALL THE ESSENTIAL FEATURES OF THE ESP8266, SUCH AS A PCB ANTENNA, CRYSTAL, UART SUPPORT FOR SPI AND I2C, AND 2M BYTES OF FLASH. BUT WHAT MAKES IT PARTICULARLY USEFUL FOR THIS THESIS IS THAT ALL THE CHIP PINS ARE MAPPED OUT, PROVIDING EASIER ACCESS TO THEM.

3.20 POWER SUPPLY AND CONSUMPTION :

As a Wi-Fi system on a chip, the ESP8266's transceiver requires a significant amount of power to function. However, it includes several excellent power management features, such as tightly integrated components that support greater optimization and increased efficiency. As a result, the ESP8266 is one of the least power-hungry chips in the Wi-Fi IC market. Despite this, demand for the ESP8266 remains higher than for items based on other wireless technologies, such as Bluetooth or ZigBee. According to the official ESP8266 datasheet, the chip's current draw can vary depending on the specific mode of operation and the voltage supplied to it. For example, in the active mode, the current draw can range from 70mA to 170mA, while in the sleep mode, it can be as low as 10uA.

Mode	Typ	Unit
Transmit 802.11b, CCK 11Mbps, P _{OUT} =+17dBm	170	mA
Transmit 802.11g, OFDM 54Mbps, P _{OUT} =+15dBm	140	mA
Transmit 802.11n, MCS7, P _{OUT} =+13dBm	120	mA
Receive 802.11b, packet length=1024byte, -80dBm	50	mA
Receive 802.11g, packet length=1024byte, -70dBm	56	mA
Receive 802.11n, packet length=1024byte, -65dBm	56	mA
Deep sleep	10	uA
Power save mode DTIM 1	1.2	mA
Power save mode DTIM 3	0.9	mA
Total shutdown	0.5	uA

Table 3.2 ESP8266EX current draw at 3.3V as listed in the official documentation from Espressif

FIG : 3.20.1 POWER SUPPLY AND CONSUMPTION

SINCE THE MOD-WIFI-REAL ESP8266-DEV ALSO HAD LEDs, CRYSTALS, CAPACITORS, AND REGISTERS IN ADDITION TO THE ESP8266EX CHIP, IT WAS DISCOVERED THAT ITS POWER CONSUMPTION WAS SUBSTANTIALLY HIGHER THAN THAT OF THE TABLE. WHILE THE MODULE WAS GETTING READY TO ACCEPT PACKETS, IT WAS DISCOVERED TO HAVE AN APPROXIMATE IDLE CURRENT OF 70 MA; THIS FIGURE SLIGHTLY INCREASED WHEN PACKETS WERE BEING RECEIVED IN 802.11N MODE. 80MA WAS NOW BEING USED FOR TRANSMISSION.

3.21 SERIAL COMMUNICATION :

The ESP8266 includes many built-in peripherals that provide a typical embedded interface with other devices. In this section, we will only cover the setup of the communication link, as the precise flow of bits to create such communication is handled automatically by the module and is therefore seen as having little immediate significance for this thesis. For data encoding and decoding, the standard UART was used in this case to communicate with the sensor. Similarly, the EM50 data logger, which is equipped with a UART of its own, can perform the same function. Baud rates are not required for serial asynchronous communication, but they are crucial to ensuring that the data is processed correctly and at regular intervals.

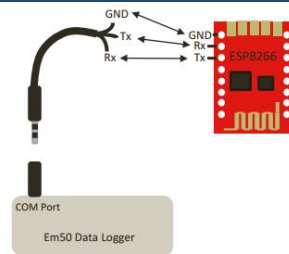


Figure 3.9 Serial communication was conducted over a 3.5mm AUX cable attaching ESP8266s Rx, Tx and GND to sensors Tx, Rx and GND respectively

Fig : 3.21.1 Serial Communication

SOFTWARE REQUIREMENTS

3.22 ARDUINO IDE:

The Arduino Integrated Development Environment (IDE) is an open-source software tool used to write and upload code to Arduino boards. It is a user-friendly platform that simplifies the process of programming microcontrollers for people with little or no programming experience.

The Arduino IDE provides a code editor with features such as syntax highlighting, auto-indentation, and code completion, making it easier to write error-free code. It also includes a serial monitor for debugging and testing, which allows users to see data sent between the Arduino board and the computer in real-time.

Furthermore, the IDE offers a wide range of libraries and examples that can be used to build projects quickly and efficiently. Users can easily upload their code to the board using a USB cable, eliminating the need for external programmers.

Overall, the Arduino IDE is an excellent tool for beginners and advanced users alike, providing a simple yet powerful platform for building and testing electronic projects.

During the process of creating a program, it is important to check for errors to ensure the program runs correctly. Additionally, before uploading the program to the chosen board, it is necessary to assemble the code.

If using an external programmer, it is possible to hold down the "shift" key to access the "Upload using Programmer" option.

The "New" option creates a new drawing, while "Open" displays a menu of all drawings in the sketchbook. Clicking on a drawing from the menu will replace the current content.

"Save" is used to save the drawing.

The "Serial Monitor" option opens the serial monitor.

Additionally, the "File," "Edit," "Sketch," "Tools," and "Help" menus have additional commands that are context-sensitive.

"New" creates a new instance of the editor with a sketch's basic framework in place, while "Open" enables loading of a sketch file from the computer drives and directories.

"Open Recent" provides a condensed list of the most recent sketches, and the "Sketchbook" option displays the current drawings within the sketchbook folder hierarchy.

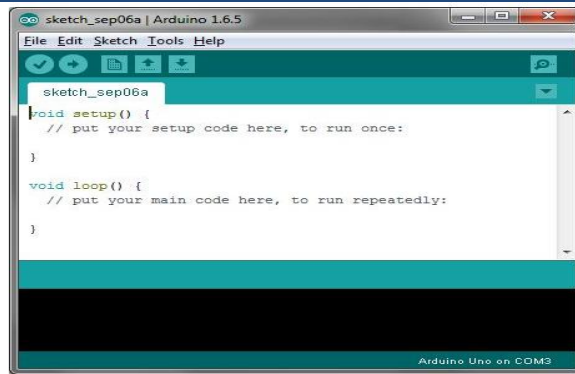
Clicking on a name opens the related sketch in a new editor instance.

- The "Examples" option in the Arduino Software (IDE) displays available examples from the IDE or library, organized in a tree format that is easy to browse by topic or library.
- The "Close" option closes the current instance of the Arduino Software.
- "Save" saves the current drawing with its current name, while "Save as..." enables saving the drawing under a different name.
- "Page Setup" opens the printing page setup window, and "Print" sends the active drawing to the printer based on the preferences set in the page setup.
- "Preferences" opens the preferences box where IDE settings and language can be changed. "Quit" shuts down all IDE windows, and any drawings that were open when the Quit option was selected will be opened by default the next time the IDE is launched.
- The "Edit" menu has options for undo/redo, cut, copy, copy for forum, and copy as HTML. Undo/redo allows you to undo or redo one or more editing steps.
- "Cut" removes the selected text from the editor and places it on the clipboard. "Copy" copies the editor's currently-selected text and adds it to the clipboard.
- "Copy for Forum" and "Copy as HTML" copy the sketch's code to the clipboard in a format suitable for forum posting, including syntax colouring.
- The "Edit" menu has several options, including "Paste," which inserts the contents of the clipboard at the cursor point.
- "Select All" highlights and selects all of the editor's contents. "Comment/Uncomment" adds or removes the comment marker at the start of each selected line.

- "Increase/Decrease Indent" moves the text one space to the right or left by adding or removing a space at the start of each selected line.
- "Find" opens the Find and Replace box, where you can specify how to search for text within the current drawing.
- "Find Next" highlights the next occurrence of the string that was selected as the search item in the Find window, depending on where the cursor is located.
- "Find Previous" does the same but finds the previous occurrence of the string.

Sketch

- *Verify/Compile* At compilation, your sketch is checked for defects; a console section will provide information about memory use for the code and variables.
- *Upload* compiles and loads the binary file through the specified Port onto the intended board.
- *Upload Using Programmer* This will replace the bootloader on the board; to restore it and resume using the USB serial interface, choose Tools > Burn Bootloader. Yet, it enables you to use the entire Flash memory for your sketch. Please take note that this instruction will NOT burn the fuses. To accomplish this, use the Tools -> Burn Bootloader command.
- *Export Compiled Binary* creates a .hex file that can be delivered to the board using other tools or archived for later use.
- *Show Sketch Folder* Opens the current sketch folder.
- *Include Library* Add a library to your drawing by beginning your code with a #include statement. The libraries listed below provide more information. You may also use the Library Manager and import new libraries from .zip files from this menu item.
- *Add File...* incorporates a source file into the sketch (it will be copied from its current location). In the sketch window, the new file displays in a new tab. By selecting the tiny triangle icon located below the serial monitor symbol on the right side of the toolbar, which is accessible by clicking, the sketch's tab menu can be used to remove files.
- Tools
 - *Auto Format* This neatly prepares your code by indenting it and aligning the opening and closing curly braces, as well as increasing the indentation of the statements inside curly braces.
 - *Archive Sketch* Save a .zip file containing a copy of the most recent sketch. A directory containing both the sketch and the archive is chosen.
 - *Fix Encoding & Reload* corrects any potential inconsistencies between the editor's char map encoding and that of other operating systems.
 - *Serial Monitor* opens the serial monitor window and begins data communication with any attached board on the presently specified Port. When the board's Reset over Serial Port Opening feature is enabled, this often resets the board.
 - *Board* Choose the board you are currently using. For descriptions of the various boards, see below.
 - *Port* This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
 - *Programmer* in place of the onboard USB-serial connection when choosing a hardware programmer to programme a board or chip. If you're burning a bootloader to a new microcontroller, you will need this even if you normally won't need it.
 - *Burn Bootloader* An Arduino board's microcontroller can be burned with a bootloader using the items in this menu. When using an Arduino or Genuino board ordinarily, this is not necessary, however it can be helpful if you buy a new ATmega microcontroller (which normally come without a bootloader).
 - Help
 - Many documentation that are included with the Arduino Software are accessible here (IDE). Without an online connection, you can still access the Getting Started, Reference, this IDE manual, and other information that are available locally.
 - *Find in Reference*
 - The only interactive feature of the Help menu is this one, which chooses the appropriate page in the local copy of the Reference for the function or command currently being used.
- **4.2 Arduino IDE: Initial Setup**
 - To configure the board and port in the Arduino IDE, follow these steps:
 - Open the Arduino IDE on your computer.
 - From the "Tools" menu, select "Board" and choose the appropriate board for your project. For example, if you are using an Arduino Uno, select "Arduino Uno."
 - Next, select the correct port from the "Tools" menu. The port should be the one your Arduino board is connected to. If you are unsure which port to select, you can check your computer's device manager to see which port the Arduino board is using.
 - Once the board and port are configured, you can upload your code to the board by clicking the "Upload" button in the Arduino IDE.
 - After uploading the code, you can monitor the output using the serial monitor in the Arduino IDE. The serial monitor can be accessed by clicking the magnifying glass icon in the top-right corner of the IDE or by pressing "Ctrl+Shift+M."



Arduino IDE Default Window

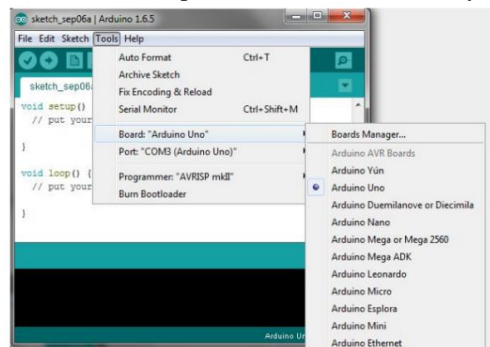
Fig : 3.23.1 Arduino IDE: Initial Setup

3.25 IDE: Board Setup :

In order to upload code to an Arduino board using the Arduino IDE, the IDE needs to know which board you are uploading to. This is done by selecting the appropriate board from the "Tools" menu.

The "Boards" menu in the Arduino IDE lists all of the Arduino boards that are currently supported by the IDE. If you are using a different board or a clone of an Arduino board, you may need to install additional board support packages in order to use it with the IDE.

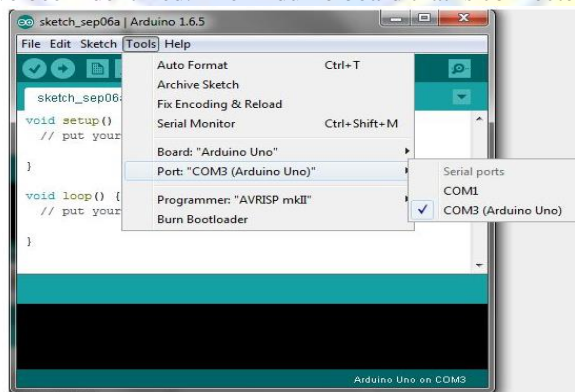
Once you have selected the appropriate board from the "Boards" menu, you can proceed to write and upload your code to the board. The Arduino IDE will automatically compile the code and upload it to the board when you click the "Upload" button.



Arduino IDE: Board Setup Procedure

Fig : 3.25.1 IDE: Board Setup

3.26 IDE: COM Port Setup : If you downloaded the Arduino IDE before connecting your Arduino board, the USB drivers should have been set up automatically. The latest Arduino IDE version ought to be able to recognise connected boards and determine which COM port they are using. From the Tools drop-down menu, select Port. All open COM ports are listed here along with the name of any Arduino Boards that are present and have been identified. The Arduino board that is connected to your PC should be chosen.




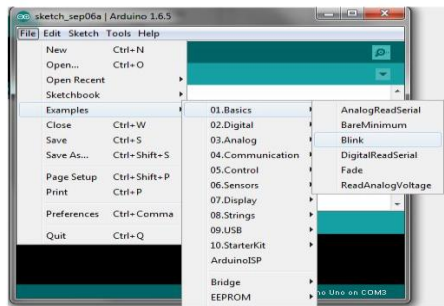
Arduino IDE: COM Port Setup

Fig : 3.26.1 IDE: COM Port Setup :

3.27 Testing Your Settings:

Blink being raised It is customary to upload the "Blink" sketch to check the board's configuration. This sketch is pre-installed in all versions of the Arduino IDE and may be found by selecting File from the pull-down menu, Examples from the list, selecting 01.Basics, and then selecting Blink. On normal Arduino boards, the "L" or "LED" surface-mounted LED is linked to digital pin 13, together with the "RX" and "TX" LEDs. Open the "Blink" drawing and use the "Upload" button in the top-left corner to upload it to the board.

Upload Button: 



Arduino IDE: Loading Blink Sketch

Fig : 3.28.1 Testing Your Settings

3.29 PROTEUS:

PCB stands for Printed Circuit Board. It is a board made of insulating material, such as fiberglass, with copper traces or conductive paths printed onto it. These conductive paths serve as the electrical connections between components on the board.

PCBs are commonly used in electronic devices, from simple toys and gadgets to complex computer systems and industrial machinery. They provide a reliable and efficient way to connect electronic components, and they can be designed and manufactured in large quantities relatively quickly and inexpensively.

PCB layout refers to the process of designing the layout or arrangement of the copper traces and other components on the board. PCB designing refers to the overall process of designing the board, including selecting the components, designing the layout, and preparing the board for manufacturing.

There are two main types of PCBs: single-sided and double-sided. Single-sided PCBs have copper traces on only one side of the board, while double-sided PCBs have copper traces on both sides. There are also multi-layer PCBs, which have several layers of copper traces sandwiched between layers of insulating material.

In addition to their use in electronic devices, PCBs are also used in a variety of other applications, such as automotive systems, medical equipment, and aerospace technology.

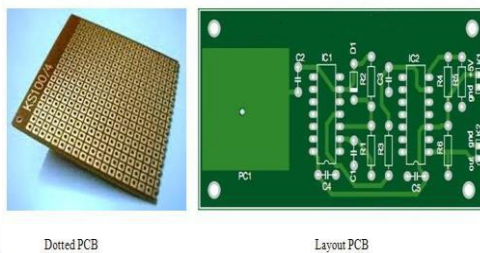


Fig : 3.29.1 PROTEUS

The workspace with interface buttons for building circuits will then display, as seen in the figure below. As visible in the workspace, the blue rectangle region must contain the full circuit.

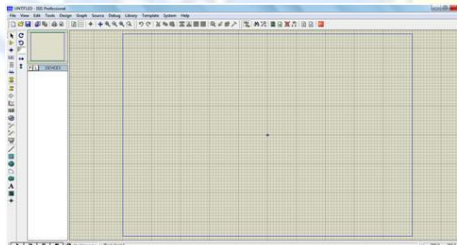


Fig : 3.29.2 PROTEUS

In the circuit diagram you provided, the 555 timer IC is used to generate a 38 kHz frequency signal. The 555 timer IC is a popular and versatile integrated circuit that can be used for a wide range of applications, including oscillators, timers, and pulse generators.

To select the components for this circuit, you will need to consider the specifications of the 555 timer IC and the other components in the circuit. The 555 timer IC has several pins that are used for different functions, so you will need to consult the datasheet for the IC to determine which pins you will need to connect to other components in the circuit.

Overall, the process of selecting components for a circuit involves researching the specifications of the components, calculating values based on the requirements of the circuit, and testing the circuit to ensure that it functions properly.

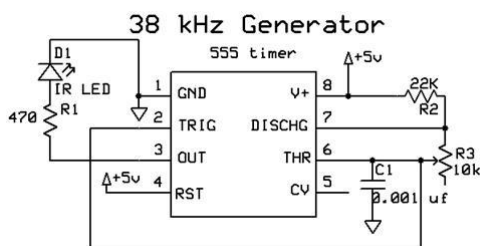


Fig : 3.29.3 PROTEUS

As an AI language model, I don't have access to any images or menus. However, I can provide guidance on selecting the components you need for the circuit based on the values you provided. For the circuit you described, you will need the following components:

- 555 timer IC
- 470 ohm resistor

- 22k ohm resistor
- 10k ohm variable resistor
- 0.001 uF capacitor
- IR LED

You can select these components from a library of electronic components, either in software like LTSpice or in a physical catalog of components. When selecting components, make sure to choose parts with the correct values and specifications for your circuit. For example, when selecting the 470 ohm resistor, choose a resistor with a tolerance and power rating that meet the requirements of your circuit. Similarly, choose an IR LED with the correct specifications for your circuit, such as the desired wavelength and viewing angle.

Once you have selected the components, you can use a software tool like LTSpice to simulate the circuit and ensure that it functions as expected. If the circuit does not work correctly, you may need to adjust the values of the components or choose different components with different specifications.

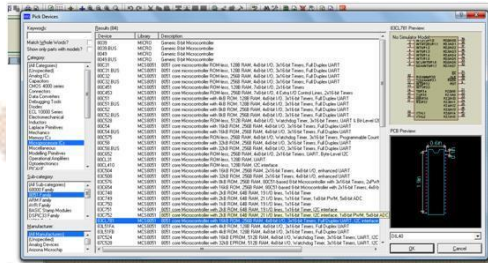


Fig : 3.29.4 PROTEUS

To choose the components, there is another method. On the workspace's left side, a tool bar is present. Either select an option from the toolbar's library or click the component mode button

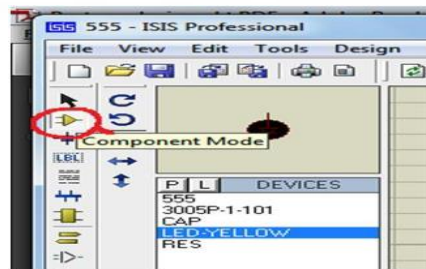


Fig : 3.29.5 PROTEUS

To add each component to the list of devices, select it from the library. When clicking on the device to change its angle, use the rotate buttons. You click there, and then the selected component is added to the work area. With the cursor at the component pin end, use the pen symbol to draw connections between all the devices that have been placed on the work area.

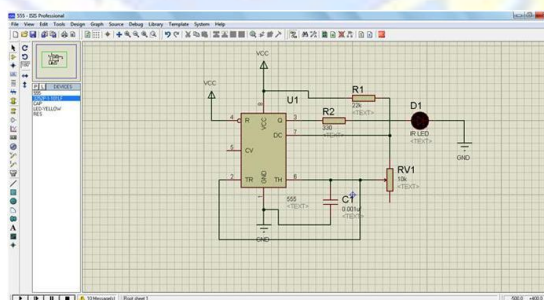


Fig : 3.29.6 PROTEUS

When the mouse pointer is above the component you want to change, click the right mouse button, which will bring up an options box. The figure below shows how that works.

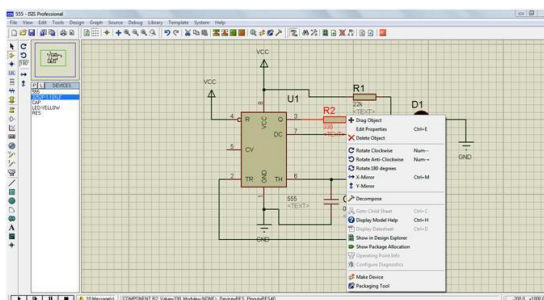


Fig : 3.29.7 PROTEUS

After finishing the design, save it with some settings and debug it. Since that this simulation is virtual, we can use the software to build the PCB layout for the required circuit and view the results digitally without actually building the circuit.

3.30 FIREBASE IOT PLOTFORM :

IoT programmes employing a real-time database powered by Firebase.

3.31 Realtime database :

The JSON format is used to store databases, which are synchronised in real time with clients. The cross-platform client is the database's main architectural design, enabling all clients to access the same resource from the Firebase server and continuously updating itself as new or updated data becomes available.

3.32 Offline work :

Even in the event of a failed internet connection, the Firebase application keeps running. The local database of Firebase will save any data that is written to the cloud. Upon resumption of internet service, client activity will be immediately updated and promptly synchronised with the server.

3.33 User authentication :

The anonymous authentication technique, GitHub, Google, Facebook, and Twitter were created by Firebase. In the form of tokens or API secret keys, this feature can be added to the application's back-end code.

3.34 Firebase Hosting :

This provides an easy method for immediately hosting a free website on the NodeJs platform. By following a few steps, we may use the free domain "WEB-APP-NAME.firebaseio.com" to transfer the full website, including the HTML, CSS, and Javascript files, to the cloud. We have the option to link the web application to the private domain or change it.

3.35 MIT APPINVENTOR :

Thanks to its vast sensor technology, simple access to a virtually infinite amount of online content, and storage for private data, the smartphone functions as an information hub in the contemporary digital era. The power of these omnipresent devices is, however, not always fully utilised by people for themselves and their communities. Even while cellphones are commonly utilised to address local issues, the bulk of smartphone users can only use their devices to consume technology.

IV. WORKING

The IoT-based engine oil level and impurities monitoring system uses a portable online spectrophotometer built into the dipstick, which is inserted into the engine's oil sump. The system works based on spectrophotometry, measuring light absorption and transmission by a sample. The online spectrophotometer has a light source emitting specific wavelengths and a detector measuring the transmitted light's intensity through the oil sample. The spectrophotometer identifies impurities by measuring the absorption of light at specific wavelengths.

To calibrate the system, a clean oil sample is used as the baseline. The dipstick is then inserted into the oil sump, and the spectrophotometer takes measurements as the oil flows through it. Any changes in light absorption indicate the presence of impurities in the oil, allowing for real-time monitoring of the oil level and impurities. This proactive maintenance extends the engine's life, and the system's internet connectivity enables remote monitoring and alerts for potential engine issues.

V. CONCLUSIONS

A CUTTING-EDGE METHOD OF CHECKING ENGINE OIL LEVELS AND IMPURITIES IS THE DIPSTICK TYPE ENGINE OIL LEVEL AND IMPURITIES MONITORING SYSTEM. THE SYSTEM IS PORTABLE, SIMPLE TO USE, AND OFFERS REAL-TIME MONITORING, WHICH MAKES IT AN INDISPENSABLE TOOL FOR ENGINE MAINTENANCE. WITH THE USE OF A PORTABLE ONLINE SPECTROPHOTOMETER, IT IS NOW FEASIBLE TO IDENTIFY ANOMALIES IN ENGINE OIL EARLY ON, PROTECTING THE ENGINE FROM MORE SERIOUS HARM.

VI. REFERENCES

- [1] T. Muthamilselvan et al., "We are Intech Open , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 % Control of a Proportional Hydraulic System," Intech open, vol. 54, no. June, pp. 713–727, 2016.
- [2] D. Perakovi and M. Peri, *Advances in Design, Simulation and Manufacturing*, vol. 1. Springer International Publishing, 2019.
- [3] B. Okonokhua, B. Ikhajagbe, G. Anoliefo, and T. Emede, "The Effects of Spent Engine Oil on Soil Properties and Growth of Maize (*Zea mays L.*)," *J. Appl. Sci. Environ. Manag.*, vol. 11, no.3, 2010.
- [4] J. A. Heredia-Cancino, M. Ramezani, and M. E. Álvarez-Ramos, "Effect of degradation on tribological performance of engine lubricants at elevated temperatures," *Tribol. Int.*, vol. 124, pp. 230–237, 2018.
- [5] J. Ma, Z. Zong, F. Guo, Y. Fei, and N. Wu, "Thermal Degradation of Aviation Synthetic Lubricating Base Oil," *Pet. Chem.*, vol. 58, no. 3, pp. 250–257, 2018.